ELSEVIER

Journal of Computational and Applied Mathematics 67 (1996) 1–14

JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS

Solution of constrained nonlinear equations in modelling the release of liquified gases

J.M. Aitchison*, N.K. Upton[1]

Applied Mathematics & Operational Research Group, Cranfield University, RMCS Shrivenham, Swindon SN6 8LA, United Kingdom

Received 8 February 1993; revised 20 September 1994

Abstract

This paper presents a mathematical model of the behaviour of a cloud formed immediately after the sudden release of a pollutant. The model is appropriate for either a release into a confined space or as the first stage of a model of release to the atmosphere.

An effective algorithm is given for determining feasible numerical solutions of the system of constrained nonlinear equations and complementarity relations which constitute the model.

Keywords: Constrained nonlinear equations; Complementarity relations; Pollution modelling

1. Introduction

The study of the accidental release of certain hazardous substances is important due to their widespread use in manufacturing industry and the detrimental effects that can be caused by such releases. The worst accident of its kind occurred at Bhopal where more than two thousand people were killed by a poisonous gas.

A low boiling point substance can be stored or transported in liquid phase at a temperature in excess of its normal boiling point by compression within a vessel at a sufficiently high pressure. Examples are ammonia and chlorine with respective boiling points of about $-33°C$ and $-35°C$. Rupture of the vessel would cause some or all of its superheated contents to be ejected in liquid phase (*fine droplet aerosol*) or vapour phase (*vapour flash*) or both. Ammonia releases have been investigated theoretically in [5], and work in this field has been reviewed in [4].

In Section 2 we present a mathematical model which describes the behaviour of a polluted cloud formed almost immediately after an instantaneous release. The model can be used to predict the

* Corresponding author. E-mail: aitchison@rmcs.cran.ac.uk.

[1] Present address: St Catharines College, Cambridge CB2 IRL, UK.

SSDI 0377-0427(94)00120-0

density and temperature of the cloud after thermal equilibrium but before the occurrence of dispersion or air entrainment. In particular, we can determine whether the resulting cloud is denser than the surrounding air, creating a greater safety hazard than a buoyant cloud.

The model includes linear and nonlinear algebraic equations subject to constraints. The numerical solution of such problems has been discussed in [8]. However, the present system also involves complementarity relations, so that the actual set of equations satisfied by the feasible solution is not known until after the solution has been determined. This requires the development of new methods if these equations are to be solved in a reliable way. In Section 3 we describe a new robust algorithm than can be applied to find numerical solutions of the model. The model and algorithm are illustrated by some results in Section 4.

2. Mathematical model

We consider the sudden release of pollutants into the atmosphere following the failure of a pressure vessel. The model will describe the release of several pollutants with differing properties into air which may already contain water vapour. The pollutants may be released in both liquid and vapour form. Following the release, the components of the resulting cloud of air and aerosol pollutant undergo changes of temperature and possibly of phase until thermal equilibrium is achieved. The aim of the model is to predict the density and the common temperature of this cloud when thermal equilibrium is attained. The model therefore describes the period before atmospheric dispersion becomes important. In particular, the model seeks to determine whether the cloud will be colder or denser than the ambient air.

Thermal equilibrium is likely to be achieved over a very short time scale and so in the development of the model we assume that the following conditions hold.

(i) Gases and vapours behave as ideal gases.
(ii) The cloud is adiabatic and at the same pressure as the surrounding air.
(iii) No chemical reactions occur.
(iv) Neither freezing nor precipitation occurs.
(v) Specific and latent heats are independent of temperature.
(vi) The volume of the aerosol is negligible.

For simplicity, the host fluid will be assumed to be in the vapour phase. Clearly, this presents no difficulty provided that interest is directed at pollutant release into the Earth's atmosphere. However, the model would require minor revision if it were to be applied in some other context where the host fluid was not air.

Assume that the release is made into an initial volume V_0 of damp air with relative humidity ϕ. Assume also that prior to the release that the air is at ambient temperature T_0 and atmospheric pressure P. Initially, this air is composed of a mass m_a of dry air and a mass m_w of water vapour. These masses can be easily calculated from V_0, ϕ, T_0 and P. The density of the ambient air, ρ_a, can then be calculated as

$$\rho_a = \frac{m_a + m_w}{V_0}.$$

Assume that there are N different chemical components in the cloud including water (if present) but excluding dry air. Initially, each component may exist in both liquid and vapour form with each phase at a prescribed temperature. The total mass of component i is labelled M_i.

When thermal equilibrium is reached, each component may have partially (or totally) evaporated or condensed. Let the ith chemical component have a final vapour mass m_i. The corresponding liquid mass can then be found from the principle of conservation of mass as $M_i - m_i$.

We can construct an energy equation relating the initial masses and temperatures of the components of the cloud, which are all given as initial data, to the unknown final masses and common temperature (see [9] for details). The energy equation is a summation of heat balance terms for each component. Following elimination of the masses of the liquid phases we get an equation of the form

$$E(T_f, \boldsymbol{m}) = 0, \tag{1}$$

where $\boldsymbol{m} = [m_1, m_2, \ldots, m_N]$ and T_f is the final common temperature of the cloud. The energy function E takes the form

$$E(T_f, \boldsymbol{m}) = \kappa_1 T_f - \sum_{i=1}^{N} [\kappa_{2,i} T_f - \kappa_{3,i}] m_i - \kappa_4, \tag{2}$$

where the positive constants κ_1, $\kappa_{2,i}$, $\kappa_{3,i}$, κ_4 depend on the thermal characteristics of the different components of the cloud and the initial conditions.

Application of the ideal gas laws to the separate vapour components of the cloud provides

$$P_a V_f = \frac{m_a}{W_a} R T_f, \tag{3}$$

$$P_i V_f = \frac{m_i}{W_i} R T_f, \quad i = 1, \ldots, N, \tag{4}$$

where P_a, W_a, $\{P_i\}$, and $\{W_i\}$ represent the partial pressures and molecular weights of the dry air and the vapour components of the pollutants. V_f is the final volume of the cloud which, in general, will be different from the initial volume of air V_0, and R is the universal gas constant. Note that to allow easy use of the gas laws, all temperatures introduced in the model are absolute temperatures, although numerical results will be expressed in degrees Celsius.

It is assumed that the total pressure of the cloud is the same as the ambient atmospheric pressure, and so Dalton's law of partial pressures gives

$$P_a + \sum_{i=1}^{N} P_i = P. \tag{5}$$

The balance between the final liquid and vapour phases of each component of the pollutant is determined from the supposition that either (i) all of the aerosol has evaporated and that the resulting vapour is not saturated, or (ii) the vapour is saturated and some airborne droplets remain.

This can be conveniently written as the following complementarity relations, where $S_i(T_f)$ is the saturated vapour pressure of component i at temperature T_f:

$$\left.\begin{aligned} &[m_i - M_i][P_i - S_i(T_f)] = 0 \\ &m_i \leqslant M_i \\ &P_i \leqslant S_i(T_f) \end{aligned}\right\} \quad i = 1, \ldots, N. \tag{6}$$

The saturated vapour pressure $S_i(T)$ of each component can be calculated from the Clausius–Clapeyron equation, see [6, p. 655], as a prescribed function of temperature which can be written as

$$S_i(T) = \beta_i \exp(\alpha_i / T), \tag{7}$$

where $\alpha_i < 0$ and $\beta_i > 0$ are known for all i.

To avoid any nonphysical solutions to the above system of equations and complementarity conditions we impose the following simple constraints on the variables:

$$\begin{aligned} &V_f \geqslant 0, \qquad P_a \geqslant 0, \\ &P_i \geqslant 0, \quad i = 1, \ldots, N. \end{aligned} \tag{8}$$

Subsequent to thermal equilibrium, the polluted cloud is described by T_f, V_f, P_a, $\{P_i\}$, and $\{m_i\}$, where $i = 1, \ldots, N$. These $(2N + 3)$ variables must be determined by solving the system of $(2N + 3)$ relations (1), (3)–(6), subject to the constraints (8). A suitable algorithm is presented in the next section.

The final density of the cloud ρ_c can be derived from its mass and volume:

$$\rho_c = \frac{m_a + \sum_{i=1}^{N} M_i}{V_f}.$$

Note that the cloud is denser than the ambient air if $\rho_c / \rho_a > 1$.

3. Numerical algorithm

3.1. Development of the algorithm

The objective is to develop an algorithm suitable for finding numerical solutions to the multiple pollutant model described in Section 2. The $(2N + 3)$ relations (1), (3)–(6) involve the $(2N + 3)$ variables T_f, V_f, P_a, $\{P_i\}$, $\{m_i\}$ where $i = 1, \ldots, N$. There are several special features of the system which are worth noting. Firstly, there are both linear and nonlinear algebraic equations in the system, and there is a repetition of the pattern of the nonlinearity in Eqs. (3) and (4). Secondly, the variables are subject to the simple inequality constraints (8). Thirdly, the complementarity relations

(6) require that either $m_i = M_i$ or $P_i = S_i(T_f)$, for $i = 1, \ldots, N$. However, it is not known precisely which of these equations are satisfied until the solution has been determined. Although it might be feasible to solve this system using a general purpose constrained nonlinear equation solver, the strategy adopted in this paper will be to construct an efficient method that exploits the special structure of the above system.

The algorithm described below involves the solution of a number of constrained linear equations, but only one unconstrained nonlinear equation. Firstly, we note that if the value of T_f is specified then Eqs. (3)–(6) can be rearranged as a constrained linear system which can be solved to give all the other unknown variables. (This is described in detail below as the inner procedure.) In particular, the vapour masses in the vector $\boldsymbol{m}$ can be written as functions of T_f. These can be substituted into the expression for E in Eq. (2), so that $E(T_f, \boldsymbol{m})$ can be effectively written as $E(T_f)$ and evaluated for any choice of T_f. Finally, Eq. (1) is solved for T_f using an iterative root-finder applied to the single nonlinear equation, described below as the outer procedure. Each iterative step in this root-finding process involves the evaluation of E for a specified value of T_f which in its turn requires the solution of a constrained linear system using the inner procedure.

Outer procedure

There are many good root-finding techniques available for finding the zeros of a single nonlinear function. For the present application we require a robust method which does not use derivatives. The function $E(T_f)$ defined by Eq. (2) is believed to be continuous and monotonically increasing, and so a bracketing method is appropriate.

A well-known bracketing method is the *regula falsi* algorithm, which is robust but only linearly convergent. Also the progress of the method is slowed once the function is either convex or concave across the bracketing interval, since one of the bracketing points is retained throughout the remaining iterations reducing the convergence rate. This will be true for most functions if the bracket is sufficiently small. Dowell and Jarratt [2] published a modified regula falsi algorithm, known as the *Illinois* method, which attempts to overcome this difficulty and is more efficient than the original regula falsi algorithm. Another variation of the algorithm is the *Pegasus* method suggested in [3]. The superiority of Pegasus over the Illinois method has been demonstrated in practice [3, 7]. Other algorithms offer higher theoretical orders of convergence at the expense of greater computational complexity [9].

After experimentation the Pegasus method [3] was found to give a satisfactory performance for the system considered in this paper. The Pegasus method requires (i) the evaluation of the function E given by Eq. (2), and (ii) an initial bracket $[T_\ell, T_u]$ that encloses the solution of the nonlinear equation (1). A reasonable practical choice for the initial interval is $[\frac{1}{2}T_0, \frac{3}{2}T_0]$, but this could be altered or reduced if a good estimate of the final temperature were available from experimental information.

Inner procedure

Each iteration of the outer procedure requires the solution of Eqs. (3)–(6) for a specified value of T_f. Therefore, suppose that T_f is given. Adding (3) and (4) and then substituting into (5)

gives

$$PV_f = \left[\frac{m_a}{W_a} + \sum_{j=1}^{N} \frac{m_j}{W_j}\right] RT_f. \tag{9}$$

Dividing (4) by (9) to eliminate V_f and T_f gives

$$\frac{P_i}{P} = \frac{m_i/W_i}{(m_a/W_a) + \sum_j (m_j/W_j)}, \quad i = 1, \ldots, N. \tag{10}$$

Throughout the remainder of this section, it will be assumed that summations begin at 1 and end at N, except where stated to the contrary, with only excluded values in the index shown explicitly.

Since T_f has been specified, $S_i(T_f)$ is simply a positive constant for each i, and so we can write $S_i = S_i(T_f)$. From the complementarity relations (6), $P_i \leqslant S_i$, and so using Eq. (10) to eliminate P_i gives

$$P \frac{m_i}{W_i} \leqslant S_i \left[\frac{m_a}{W_a} + \sum_j \frac{m_j}{W_j}\right], \quad i = 1, \ldots, N.$$

Gathering the terms in m_i/W_i and dividing by S_i, this can be written as

$$\left[\frac{P}{S_i} - 1\right] \frac{m_i}{W_i} - \sum_{j \neq i} \frac{m_j}{W_j} \leqslant \frac{m_a}{W_a}, \quad i = 1, \ldots, N. \tag{11}$$

Relation (11), together with the constraint $m_i \leqslant M_i$ from the complementarity relations (6), can now be solved for $\boldsymbol{m} = [m_1, m_2, \ldots, m_N]^T$ as follows.

For $i = 1, \ldots, N$, define x_i by

$$x_i = \frac{M_i - m_i}{W_i}. \tag{12}$$

Then (11) becomes

$$\left[\frac{P}{S_i} - 1\right] x_i - \sum_{j \neq i} x_j \geqslant -\frac{m_a}{W_a} - \sum_j \frac{M_j}{W_j} + \frac{P}{S_i} \frac{M_i}{W_i}, \quad i = 1, \ldots, N. \tag{13}$$

Let

$$a_i = \frac{P}{S_i} - 1, \quad i = 1, \ldots, N,$$

$$b_i = -\frac{m_a}{W_a} - \sum_j \frac{M_j}{W_j} + \frac{P}{S_i} \frac{M_i}{W_i}, \quad i = 1, \ldots, N,$$

where we note that b_i is a constant. We also observe from (12) that $x_i \geqslant 0$ if and only if $m_i \leqslant M_i$. Therefore, the complementarity conditions (6) can be expressed as

$$\begin{aligned} &\boldsymbol{x}^{\mathrm{T}}[\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}] = 0, \\ &\boldsymbol{x} \geqslant \boldsymbol{0}, \\ &\boldsymbol{A}\boldsymbol{x} \geqslant \boldsymbol{b}, \end{aligned} \tag{14}$$

where $\boldsymbol{x} = [x_1, x_2, \ldots, x_N]^{\mathrm{T}}$, $\boldsymbol{b} = [b_1, b_2, \ldots, b_N]^{\mathrm{T}}$, and

$$\boldsymbol{A} = \begin{bmatrix} a_1 & -1 & -1 & \cdots & -1 \\ -1 & a_2 & -1 & & \\ -1 & -1 & a_3 & & \\ & & & & \vdots \\ & & & \ddots & \\ \vdots & & & & \\ & & & a_{N-1} & -1 \\ -1 & & \cdots & -1 & a_N \end{bmatrix}. \tag{15}$$

Cryer [1] proposed the use of modified versions of classical iterative techniques to solve constrained systems of linear equations such as (14). The *modified Gauss–Seidel* method is defined for the system specified by (14) and (15) as follows:

$$\begin{aligned} \bar{x}_i^{(k)} &= \frac{1}{a_i}\left[b_i + \sum_{j<i} x_j^{(k)} + \sum_{j>i} x_j^{(k-1)} \right], \\ x_i^{(k)} &= \max[\bar{x}_i^{(k)}, 0], \end{aligned}$$

where the superscript k is the iteration number. An initial vector is required to start the iteration which can be taken as $\boldsymbol{x}^{(0)} = \boldsymbol{0}$. This iteration then produces a sequence $\boldsymbol{x}^{(k)}$ which, under suitable conditions, converges to the required solution $\boldsymbol{x}$. The question of theoretical and practical convergence will be discussed later in this paper.

If $S_i > P$ for any particular value of i, then the constraint $P_i = S_i$ cannot be satisfied and so the complementary relation $m_i = M_i$ must hold. In such cases the value of $x_i^{(k)}$ is set to zero throughout the inner iteration for that particular value of T_f.

Once $\boldsymbol{x}$ has been determined, the vector of vapour masses $\boldsymbol{m}$ can be recovered from (12), and substituted into Eq. (2) for use in the outer root-finding procedure. The values of V_f, $\{P_i\}$ and P_a are not needed for the substitution into the energy function. They can be calculated after the outer procedure has converged and a final value of T_f has been found from Eqs. (9), (10) and (5).

3.2. Convergence of the algorithm

Outer procedure

Fig. 1 shows the graph of $E(T_f)$ as used in the outer iteration for one of the test problems described in Section 4. This function was determined experimentally for several different cases and

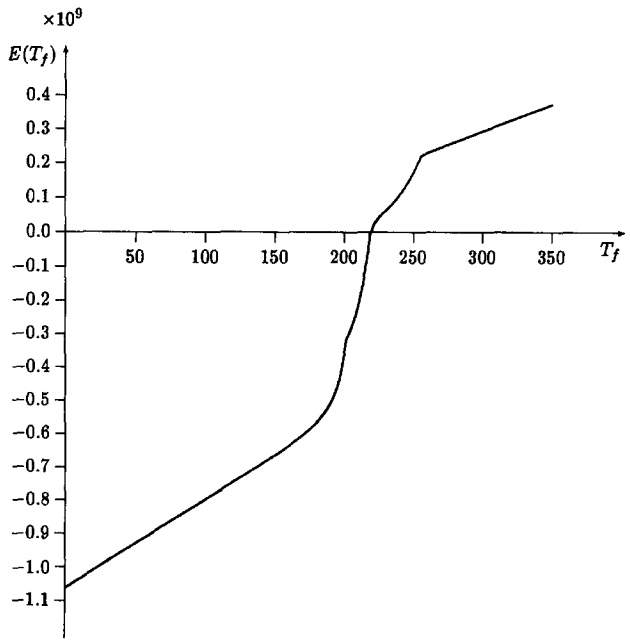

Fig. 1. Energy function $E(T_f)$ from test 2.

the behaviour shown in Fig. 1 is typical. The figure was produced by setting T_f to a series of values, solving the constrained *linear* system for each value of T_f in turn thus calculating a corresponding value of $\boldsymbol{m}$, and finally evaluating $E(T_f, \boldsymbol{m})$ using Eq. (2).

It appears that the derivative of $E(T_f)$ is discontinuous—there is at least one kink in E, at about $T_f = 250\,\mathrm{K}$, and probably another at about 200 K. This lack of smoothness is hardly surprising since the calculation of $\boldsymbol{m}$ involves the imposition of constraints. These kinks seem to arise as one or more components move between saturation and nonsaturation. However, the function does appear to be continuous and monotone. If this is true then the Pegasus iteration will converge provided $E(T_\ell)E(T_u) < 0$.

Inner procedure

For unconstrained linear problems of the form $\boldsymbol{Ax} = \boldsymbol{b}$, it is well known that a sufficient condition for the convergence of the standard Gauss–Seidel iteration is that the symmetric matrix $\boldsymbol{A}$ should be positive definite. Cryer [1] extended this result to the application of the modified Gauss–Seidel iteration to constrained problems of the form (14), although an example will be given later in this paper to show that the condition is not necessary.

Nevertheless, it is clearly worth testing the conditions under which the matrix A used in this particular problem is positive definite. This is provided by the following theorem.

Theorem 1. *The matrix $\boldsymbol{A}$ is positive definite if and only if* $P - \sum_{i=1}^{N} S_i > 0$.

Proof. Rewrite (15) as

$$A = \begin{bmatrix} c_1 - 1 & -1 & \cdots & -1 \\ -1 & c_2 - 1 & & \\ & & & \vdots \\ & & \ddots & \\ \vdots & & & \\ & & c_{N-1} - 1 & -1 \\ -1 & \cdots & -1 & c_N - 1 \end{bmatrix}, \tag{16}$$

where $c_i = a_i + 1 = P/S_i > 0$. Necessary and sufficient conditions for a matrix to be positive definite are that each leading submatrix has positive determinant (Sylvester's conditions). Using elementary row and column operations, it is straightforward to show that the these determinants are

$$|A_1| = c_1 - 1,$$

$$|A_r| = \prod c_j - \sum_{i=1}^{r} \left[\prod_{j \neq i} c_j \right], \quad r = 2, \ldots, N,$$

where products run over j from 1 or r inclusive, apart from where an exception is indicated explicitly. Details of these calculations are given in [9].

The determinant $|A_r|$ can be more conveniently written as

$$|A_r| = \prod c_j \left\{ 1 - \sum_{i=1}^{r} \frac{1}{c_i} \right\}, \quad r = 2, \ldots, N$$

where it can be seen that $|A_r| > 0$ if and only if

$$\sum_{i=1}^{r} \frac{1}{c_i} < 1.$$

But, $c_i > 0$, and so

$$\sum_{i=1}^{r} \frac{1}{c_i} \leqslant \sum_{i=1}^{N} \frac{1}{c_i}$$

for any $r \leqslant N$. Therefore, a necessary and sufficient for A to be positive definite is

$$\sum_{i=1}^{N} \frac{1}{c_i} < 1,$$

which, in the original notation, becomes

$$P - \sum_{i=1}^{N} S_i > 0. \qquad \square$$

Theorem 1 and Cryer's extension [1], described above, lead directly to the sufficient condition for the convergence of the modified Gauss–Seidel method for the given problem.

Theorem 2. *The modified Gauss–Seidel method for the system defined by Eqs.* (14) *and* (15) *is guaranteed to converge if* $P - \sum_{i=1}^{N} S_i > 0$.

Complete algorithm

Consider the value of $T = T_z$ where T_z is defined by

$$P - \sum_{i=1}^{N} S_i(T_z) = 0.$$

Then since each of the functions $S_i(T)$ is an increasing function of T, we have

$$P - \sum_{i=1}^{N} S_i(T) > 0 \quad \text{for } T < T_z$$

guaranteeing the convergence of the inner iteration. It is conjectured that the full Pegasus–Gauss–Seidel algorithm will converge using the initial interval $(0, T_z)$ provided $E(T_z) > 0$, but the proof depends on E being monotonically increasing on $(0, T_z)$. This has been observed in individual cases, but not proved in general.

4. Numerical results

The model has been applied to a wide range of chemical substances, including butane, propane, chlorine, and various refrigerants. Many practical situations involve the release of a single pollutant into air containing water vapour. A sample set of results from a simple numerical experiment of this type is shown in Table 1. A 10 kg liquid mass of refrigerant 11 (R11) has been released at a temperature of 50°C into 100 m^3 of air at an ambient temperature of 20°C. The remaining input parameter, namely relative humidity, varied between 0% and 100%. The results of five experiments are shown. Water and R11 are labelled, respectively, 1 and 2. Temperature is expressed in °C, volume in m^3, pressure in kPa, mass in kg and density in kg m^{-3}. The full set of input data necessary for these calculations is given in the appendix.

The effects of increasing relative humidity, while holding the other input parameters constant, can be seen in Table 1. Then some of the water vapour has condensed and, given that water has a comparatively high latent heat, the consequent release of that heat has produced a significantly higher final temperature. Note that the cloud is not saturated with refrigerant in any of the five experiments, and so the variable m_2 remains at 10 kg throughout. This cannot be formally predicted in advance, but retrospectively we note that the R11 vapour has a high saturated vapour pressure at the relevant temperature, and so condensation would only occur if a much greater mass of pollutant were released into the same volume of air.

Table 1
Numerical solutions for R11

Humidity	ϕ	0%	25%	50%	75%	100%
Final temperature	T_f	7.66	7.62	8.30	12.14	15.54
Final volume	V_f	97.47	97.45	97.65	98.75	99.73
Pressure of dry air	P_a	99.58	99.07	98.60	98.31	97.99
Mass of water vapour	m_1	—	0.39	0.74	0.95	1.18
Mass of liquid water	$M_1 - m_1$	—	0.00	0.03	0.21	0.36
Pressure of water vapour	P_1	—	0.51	0.98[a]	1.27[a]	1.58[a]
Mass of R11 vapour	m_2	10.00	10.00	10.00	10.00	10.00
Mass of liquid R11	$M_2 - m_2$	0.00	0.00	0.00	0.00	0.00
Pressure of R11 vapour	P_2	1.74	1.74	1.74	1.75	1.75
SVP of water vapour	$S_1(T_f)$	—	0.94	0.98	1.27	1.58
SVP of R11 vapour	$S_2(T_f)$	56.94	56.86	58.33	67.24	76.03
Density of ambient air	ρ_a	1.1889	1.1867	1.1844	1.1821	1.1799
Density of cloud	ρ_c	1.3224	1.3203	1.3153	1.2983	1.2834
Relative density	ρ_c/ρ_a	1.1123	1.1126	1.1105	1.0983	1.0877

[a]Cloud is saturated with water vapour if $\phi = 50\%$, 75%, or 100%.

The last row of Table 1 shows the density of the polluted cloud relative to the density of ambient air, which is seen to be greater than one in all cases. Therefore, the results suggest that if an accident of this sort were to occur then the resultant cloud would be negatively buoyant. Since the molecular weight of R11 is greater than that of air this result is not unexpected. However, calculations with other chemical substances which have a molecular weight lighter than that of air have also produced dense clouds (see [9] for details).

Calculations involving examples of the type described above seem to present no numerical difficulties, but to test the algorithm thoroughly it is desirable to consider a situation where there are several different chemical components in the cloud. It is difficult to collect data from experiments for such cases and so artificial tests have been devised in order to test the convergence of the algorithm.

Table 2 shows the results of these tests where the release of a number of pollutants in both vapour and liquid phase into air is simulated. In tests 1 and 2 there are, respectively, 7 and 9 pollutants released into moist air. In test 3, a mixture of 3 pollutants in both phases is released into dry air.

The two-stage algorithm was described in Section 3.1 specifically with respect to the use of the Pegasus method and the modified Gauss–Seidel method for the outer and inner iterations. However, the underlying algorithm has been implemented using several other equation solvers. Results showing computational costs are summarized in Table 2. For these comparison purposes, the other iteration was considered to have converged if the absolute error between successive approximations to the root was less than 10^{-8}. The initial interval was set as $[T_\ell, T_u] = [\frac{1}{2}T_0, \frac{3}{2}T_0]$.

The chosen method for the outer iteration, the Pegasus method, was compared with the standard regula falsi method and the Illinois method. On the whole, the number of function

Table 2
Number of iterations needed by two-stage algorithm

Test	1		2		3	
Components (N)	8		10		3	
Iterations	Outer	Inner	Outer	Inner	Outer	Inner
Regula falsi-MJ	16	70	13	72	20	40
Regula falsi-MGS		51		98		40
Illinois-MJ	16	92	16	143	13	26
Illinois-MGS		65		99		26
Pegasus-MJ	14	74	14	125	12	24
Pegasus-MGS		53		88		24

Table 3
Final masses and pressures

Substance	i	$M_\ell^{(i)}$	$M_v^{(i)}$	$P_v^{(i)}$	$S^{(i)}(T_f)$
R114	1	97.72	402.28	17.585	17.585
Chlorine	2	0.00	500.00	52.681	90.582

evaluations needed by each method reflects the known asymptotic efficiency results for these methods. However, in test 2 regula falsi needed less evaluations than Pegasus. This is an atypical result, although it does show that the theoretical superiority of a method does not translate into a practical benefit on every occasion. The modified Jacobi method was tried as an alternative for the inner iteration, but, in general, modified Gauss–Seidel required less or the same number of inner iterations as modified Jacobi, again with the exception of the regula falsi scheme in test 2. In general, the results in Table 2 are not very surprising, and confirm the known separate convergence properties of the algorithms.

Finally, we consider an example which demonstrates the limitations of the convergence analysis in Section 3.2. Suppose 500 kg of liquid R114 and 500 kg of liquid chlorine at 160°C are ejected into 100 m^3 of dry air at an ambient temperature of 20°C. Assume that no vapour is released. The calculated final scalar variables are $T_f = -37.17$°C, $V_f = 262.62\ m^3$ and $P_a = 31.06$ kPa. The vapour masses and partial pressures are listed in Table 3.

Note that $S_1 < P$ and $S_2 < P$, but that $P - [S_1 + S_2] < 0$ so, by Theorem 1, A is not positive definite. For this simple example, this implication can be checked by inspection of the matrix A at the final temperature T_f:

$$A = \begin{bmatrix} 4.762 & -1 \\ -1 & 0.119 \end{bmatrix},$$

which has one positive and one negative eigenvalue. Therefore, the matrix is not positive definite, but nevertheless the modified Gauss–Seidel method converges showing that the positive-definite condition is not a necessary condition for convergence.

5. Conclusions

The algorithm developed in Section 3 has been shown to converge over a wide range of initial conditions and thermal parameters. The algorithm was designed to fit the particular requirements of the model developed in Section 2, and to provide a robust implementation of that model. This has demonstrated the fact that standard numerical algorithms cannot necessarily deal with the particular properties of the equations and inequalities associated with the mathematical model of a physical system.

It follows, therefore, that the precise algorithm developed here cannot be immediately generalized. However the ideas used in the development of the algorithm can certainly be applied to other nonstandard algebraic systems and the algorithm itself could be applied to other problems with a similar pattern of nonlinearities and complementarity relations.

Appendix

The numerical experiment described in Section 4 considers the release of R11 into damp air. The following values of parameters are used in the governing equations where component 1 represents water and component 2 represents R11. The parameters are given in the units quoted in the main text.

The damp air has an initial volume $V_0 = 100$ and temperature $T_0 = 293.16$. Atmospheric pressure is taken to be $P = 101\,325$ and the universal gas constant is $R = 8314.3$. The initial masses of dry air (m_a) and water vapour (M_1) depend on the relative humidity (ϕ) and are given in Table 4. The energy function E given in Eq. (2) involves the constants κ_1, $\kappa_{2,i}$, $\kappa_{3,i}$, κ_4 which depend on the

Table 4
Data for R11 experiment
(a)

ϕ	0%	25%	50%	75%	100%
m_a	118.892	118.279	117.666	117.052	116.439
M_1	0	0.386	0.772	1.158	1.545
κ_1	1.28305×10^5	1.29310×10^5	1.30315×10^5	1.31320×10^5	1.32325×10^5
κ_4	3.78750×10^7	3.92267×10^7	4.05783×10^7	4.19300×10^7	4.32817×10^7

(b)

i	1	2
$\kappa_{2,i}$	3595	276
$\kappa_{3,i}$	3.79151×10^6	2.61966×10^5
α_i	−5307.06	−2973.98
β_i	1.52135×10^{11}	2.26350×10^9

thermal characteristics of the different components of the cloud and the initial conditions. The constants κ_1 and κ_4 depend on the mass of water vapour present and so are dependent on the initial relative humidity. The relevant values are given in Table 4. The constants $\kappa_{2,i}$ and $\kappa_{3,i}$ are independent of initial conditions but must be given for each component, where $i = 1$ corresponds to water and $i = 2$ to R11. Similarly, the constants α_i and β_i in Eq. (7) must be given for each component. Both sets of constants are given in Table 4(b). The relevant molecular weights are $W_a = 28.6$, $W_1 = 18.01$ and $W_2 = 137.37$.

References

[1] C.W. Cryer, The solution of a quadratic programming problem using systematic overrelaxation, *SIAM J. Control* **9** (1971) 385–392.
[2] M. Dowell and P. Jarratt, A modified regula falsi method for computing the root of an equation, *BIT* **11** (1971) 168–174.
[3] M. Dowell and P. Jarratt, The "Pegasus" method for computing the root of an equation, *BIT* **12** (1972) 503–508.
[4] R.F. Griffiths and G.D. Kaiser, Production of dense gas mixtures from ammonia releases — a review, *J. Hazard. Mater.* **6** (1982) 197–212.
[5] S.R. Haddock and R.J. Williams, The density of an ammonia cloud in the early stages of atmospheric dispersion, *J. Chem. Tech. Biotechnol.* **29** (1979) 655–672.
[6] F.F. Huang, *Engineering Thermodynamics* (Macmillan, New York, 1988).
[7] R.F. King, Methods without secant steps for finding a bracketed root, *Computing* **17** (1976) 49–57.
[8] M. Shacham, Numerical solution of constrained nonlinear algebraic equations, *Int. J. Numer. Methods* **23** (1986) 1455–1481.
[9] N.K. Upton, Algorithmic solution of air-pollutant cloud models, Ph.D. Thesis, RMCS (Cranfield), 1991.